ADA047027

RESEARCH
REPORT

D D C

NOV 23 1977

F

Industrial & Systems
Engineering Department
University of Florida
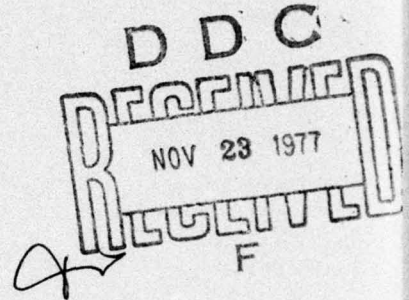Gainesville, FL. 32611

AD No.

DDC FILE COPY

A FIELD GUIDE TO IDENTIFYING
NETWORK FLOW AND MATCHING PROBLEMS

Research Report No. 77-12

by

John J. Bartholdi III
H. Donald Ratliff

September 1977

Department of Industrial and Systems Engineering
University of Florida
Gainesville, Florida   32611

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE	READ INSTRUCTIONS BEFORE COMPLETING FORM	
1. REPORT NUMBER RR-77-12	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER Research rept.
4. TITLE (and Subtitle) A Field Guide to Identifying Network Flow and Matching Problems.	5. TYPE OF REPORT & PERIOD COVERED Technical	
	6. PERFORMING ORG. REPORT NUMBER 77-12	
7. AUTHOR(s) John J. Bartholdi, III H. Donald Ratliff	8. CONTRACT OR GRANT NUMBER(s) N00014-76-C-0096	
9. PERFORMING ORGANIZATION NAME AND ADDRESS Industrial and Systems Engineering University of Florida Gainesville, Florida 32611	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 20061102A14D Rsch in Appl of Applied Math.	
11. CONTROLLING OFFICE NAME AND ADDRESS Office of Naval Research Arlington, VA	12. REPORT DATE September, 1977	
	13. NUMBER OF PAGES 37	
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office) 38 p.	15. SECURITY CLASS. (of this report) Unclassified	
	15a. DECLASSIFICATION/DOWNGRADING SCHEDULE N/A	

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from report)

N/A

D D C
NOV 23 1977
RECEIVED
F

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

networks

integer programming

matching

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

In an important but apparently little-known result, M. Iri has given necessary and sufficient conditions for a linear program to be solvable as a network flow problem. For tutorial purposes we recapitulate his result, though from a different perspective. Then in the same spirit we characterize linear programs that are solvable by the matching algorithm.

DD FORM 1473 EDITION OF 1 NOV 65 IS OBSOLETE
1 JAN 73

404 399

ABSTRACT

In an important but apparently little-known result,
M. Iri has given necessary and sufficient conditions for a
linear program to be solvable as a network flow problem.
For tutorial purposes we recapitulate his result, though
from a different perspective. Then in the same spirit we
characterize linear programs that are solvable by the
matching algorithm.

Table of Contents

1. Introduction

Since we have no solution technique of guaranteed efficiency for a general integer program, one typically attempts to solve a given problem by exploiting special structure. Most optimistically, one hopes to recognize the problem as amenable to one of the special algorithms of guaranteed efficiency, such as the minimum cost network flow algorithm [8] or the maximum b-matching algorithm [6, 7]. Indeed, many papers have "solved" problems by revealing their hidden network flow or matching structure (e.g., Bratley et al. [2], Dantzig and Fulkerson [3], de Werra [4], Dorsey et al. [5], Fujii et al. [9], Love and Vemuganti [18], Veinott and Wagner [19]). The problems are considered solved because they are "equivalent" to network flow or matching models, which are themselves efficiently solvable. We formalize the notion of equivalence and develop conditions to aid in recognizing it.

2. Equivalence of Linear Programs

Consider the linear program

1

$$\max \bar{c}\bar{x}$$
$$\text{s.t. } A\bar{x} \leq \bar{b} \tag{2.1}$$
$$\bar{x} \geq \bar{0}$$

which we may write with the addition of slack variables as

$$\max \bar{c}\bar{x}$$
$$\text{s.t. } [A, I]\,\bar{x} = \bar{b} \tag{2.2}$$
$$\bar{x} \geq \bar{0}$$

Now if T is a nonsingular matrix, the transformed problem

$$\max \bar{c}\bar{x}$$
$$\text{s.t. } [TA, T]\,\bar{x} = T\bar{b} \tag{2.3}$$
$$\bar{x} \geq \bar{0}$$

is equivalent to (2.2) in the sense that $\bar{x}$ is feasible to (2.2) iff $\bar{x}$ is feasible to (2.3). In the same way the integer-constrained versions of (2.2) and (2.3) are equivalent. We have changed merely the representation and not the essential structure of the problem. Thus solving one version of the problem solves the other. This suggests that we identify conditions on (2.2) that enable us to transform it to an equivalent network flow or matching problem.

3. Network Flows in Disguise

In an important series of papers [13-15], M. Iri has explored conditions under which a linear program may be solved by the network flow algorithm. Unfortunately his results

2

seem to be little known in the MS/IE/OR community. This is
due in part to the limited availability of the journal which
he favored. Also his work is couched in the special ter-
minology of electrical engineering, which might have limited
its accessibility. Recently some of his results were inde-
pendently rediscovered by the authors. Our approach, however,
represents a slightly different viewpoint, one which we think
may be of more immediate intuitive appeal. In the interest
of sharing Iri's results, the following recapitulation, re-
flecting our particular perspective, is offered.

Terminology

As in Johnson [16], a graph with directed arcs (i.e.,
a head at one end and a tail at the other) is a network.
A graph having only arcs which are undirected will be re-
ferred to simply as a graph. As in Edmonds et al. [7], un-
directed arcs may be imagined to have a tail at each end and
may be considered "inner-directed." A path v_0, $\bar{a}_1$, v_1, $\bar{a}_2$,
..., v_{n-1}, $\bar{a}_n$, v_n is an alternating sequence of vertices v_j
and arcs $\bar{a}_j$ where each arc $\bar{a}_j$ has one end incident to v_{j-1}
and the other end incident to v_j. For a network, a directed
path or dipath is a path with each arc $\bar{a}_j$ directed from v_{j-1}
to v_j. A cycle is a path with $v_0 = v_n$. A tree is a connected
network or graph with no cycles (whether a network or a graph will
be plain from the context). A bl6om is a graph with exactly
one cycle, that cycle consisting of an odd number of arcs.
Finally, due to the unique correspondence between a network

3

or graph G and its node-arc incidence matrix, we will refer to the matrix as G also. Again, context will identify G as a network/graph or a matrix.

Now consider the integer linear program

$$\max \bar{c}\bar{x}$$
$$\text{s.t. } A\bar{x} \leq \bar{b} \tag{3.1}$$
$$\bar{x} \leq 0, \text{ integer}$$

where A is an m x n matrix whose every entry is 0, ±1 (this may be generalized to account for scaling, as in Iri [13], but is not important to the development). Furthermore the entries of $\bar{b}$ and $\bar{c}$ are assumed to be integer.

Ford and Fulkerson [8], the network flow algorithm applies to a linear integer program which satisfies

Requirement 1: All data are integer-valued, and

Requirement 2: The constraint matrix is the node-arc incidence matrix of a network; that is, each column has no more than two nonzero entries, one a +1 and one a -1. We refer to such a matrix as a network matrix.

We seek conditions under which for some T the problem

$$\max \bar{c}\bar{x}$$
$$\text{s.t. } [TA, T]\bar{x} = T\bar{b}, \text{ T nonsingular} \tag{3.2}$$
$$\bar{x} \geq \bar{0}, \text{ integer}$$

(an equivalent form of (3.1)), satisfies Requirements 1 and 2. For (3.2) to satisfy Requirement 2, it is clear that T in particular must be a network matrix. In addition we have

4

Lemma 3.1: A network matrix is nonsingular <u>iff</u> it is the full row-rank matrix of a tree (e.g., Johnson [16]). Hence, T must be a <u>tree</u> <u>matrix</u> (i.e., the full row-rank matrix of a tree).

As an illustration, consider the tree matrix in Table 1.

$$
T = \begin{array}{c c} & \begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \end{array} \\ \begin{array}{c} A \\ B \\ C \\ D \\ E \end{array} & \left[\begin{array}{ccccc} 1 & 0 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & -1 & -1 & 1 & 1 \\ 0 & 0 & 0 & -1 & 0 \end{array} \right] \end{array}
$$

Table 1: Illustration of a Tree Matrix.

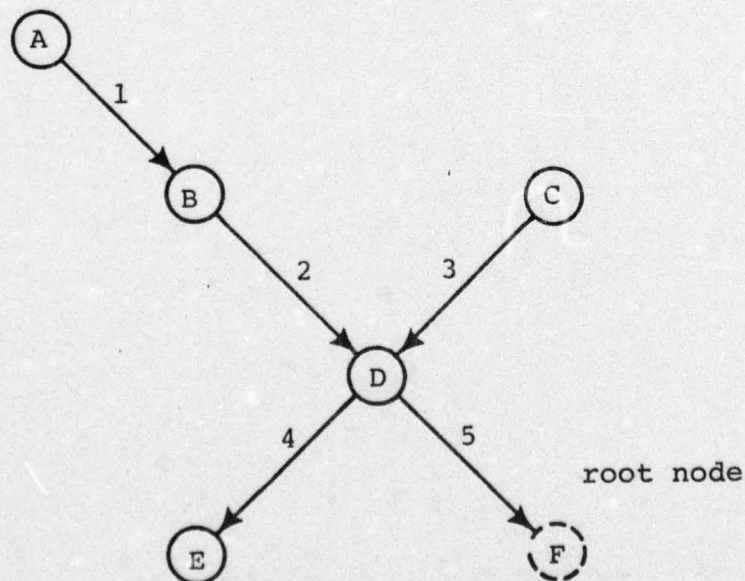This matrix corresponds to the network in Figure 1.

Figure 1: Tree Corresponding to the Matrix in Table 1.

5

Observe that a tree matrix has m columns, corresponding to m arcs of the tree. Similarly, its m rows correspond to m nodes of the tree. The $(m+1)^{st}$ node, or <u>root node</u>, is implicitly given by minus the sum of the other rows of the matrix [16].

Now to analyze TA we shall ascribe a natural network-related interpretation to A. Since T is a tree matrix, the columns correspond to the arcs of the tree T. We may imagine that through matrix multiplication this correspondence is inherited by the rows of A (i.e., each row of A corresponds to an arc of the tree T). It is then natural to interpret each column of A as follows:

(i) $a_{ij} = +1$ identifies arc i in T as a member of the set $\bar{a}_j$.

(ii) $a_{ij} = -1$ identifies arc i in T as a member of the set $\bar{a}_j$ but with its direction reversed.

(iii) $a_{ij} = 0$ indicates that arc i in T is not a member of the set $\bar{a}_j$.

For example, with regard to the matrix T in Table 1, column $\bar{a}_j = [+1, +1, 0, -1, 0]^t$ identifies as members of $\bar{a}_j$ the arcs shown in Figure 2.
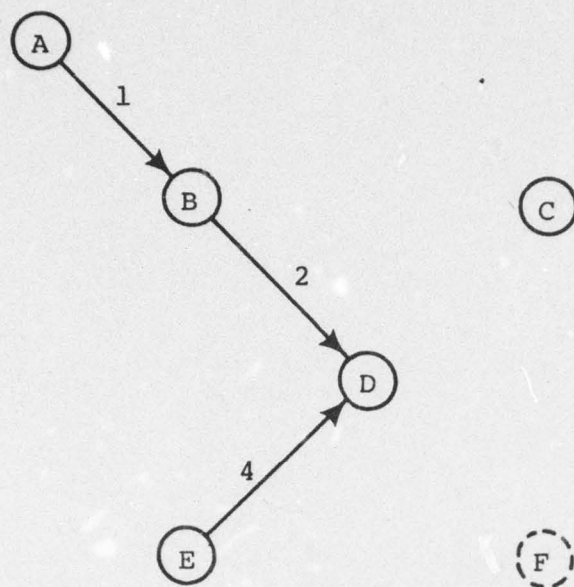
Figure 2: Arcs Identified with Respect to Matrix T in
Table 1 by the Column $\bar{a}_j = [+1, +1, 0, -1, 0]^t$.

With this interpretation imputed to A, consider the
product TA. As a network matrix, each row $\bar{t}_i$ of T gives
the degrees of incidence of arcs of T at node i; that is,

(i) $t_{ij} = 1$ <u>iff</u> the tail of arc j is incident at
node i.

(ii) $t_{ij} = -1$ <u>iff</u> the head of arc j is incident at
node it.

(iii) $t_{ij} = 0$ <u>iff</u> arc j is not incident at node 1.

Therefore the vector product $\bar{t}_i\bar{a}_j$ gives the <u>net</u> <u>degree</u> of
node i in the set of arcs $\bar{a}_j$, and $T\bar{a}_j$ is the vector of net
degrees of the nodes of T (excluding, of course, the root
node) with respect to the arcs in the set $\bar{a}_j$. As an illus-
tration, again consider the column $\bar{a}_j = [+1, +1, 0, -1, 0]^t$

and the matrix T given in Table 1. $T\bar{a}_j = [+1, 0, 0, -2, +1]^t$, which gives the net degrees of the nodes A, B, C, D, and E of Figure 2.

Now in order for TA to be a network matrix, it is necessary that no column $T\bar{a}_j$ have more than two nonzero elements. We will argue that as a consequence, the arcs of $\bar{a}_j$ must form a path in T. Clearly no more than two nodes of T, plus possibly the root node, can have nonzero net degrees in $\bar{a}_j$. Thus no more than two nodes of T, plus possibly the root node, can have an odd number of arcs of $\bar{a}_j$ incident to them. But for any graph, the number of nodes of odd degree is even [12]. Therefore no more than two nodes of T, including the root node, can have nonzero net degree in $\bar{a}_j$. Hence the arcs of $\bar{a}_j$ must form a path [12]. In particular $\bar{a}_j$ must form a simple path, i.e., one without cycles, since T is without cycles.

Now we argue that the path in T specified by $\bar{a}_j$ must possess additional special structure. First assume the path $\bar{a}_j$ does not contain the root node. Then if the net degree of the initial node is +1, that of the terminal node must be -1, and those of interior nodes must be 0. But then $\bar{a}_j$ orients the arcs of the path to form a dipath, i.e., a sequence of arcs which meet head to tail. Similarly, if the initial node is of net degree -1, $\bar{a}_j$ must still be a dipath.

Since $\bar{a}_j$ forms a simple path, if it includes the root node, it is divided into two parts by the root node. By the above argument, each part is a dipath. But the dipaths are aligned, since otherwise the net degrees of the initial and terminal nodes of $\bar{a}_j$ are the same. Then, since the two dipaths are aligned, $\bar{a}_j$ itself must be a dipath.

Since each column of A chooses and orients arcs of T to form a dipath, A must be an arc-dipath incidence matrix for a tree T. Finally we note that since T is an integer matrix, all data of problem (3.2) are integer, so that Requirement 1 is met. We have therefore

Theorem 3.1: Problem (3.2) is a network flow problem <u>iff</u> there exists some tree T for which A is an arc-dipath incidence matrix.
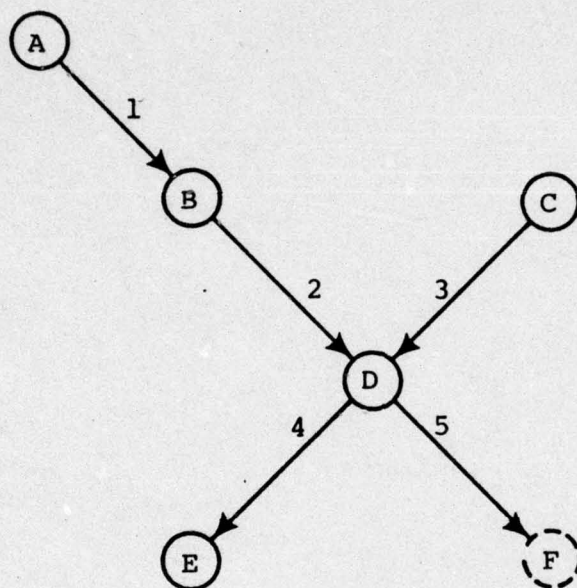
Proof: Since each step of the argument can be made "if-and-only-if," the conclusion holds.

Q.E.D.

Consider the following instance of problem (3.1),

$$\max \bar{c}\bar{x}$$

$$\text{s.t.} \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array}
\left[\begin{array}{cccc|ccccc}
1 & -1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
1 & -1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
-1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 \\
0 & -1 & 0 & -1 & 0 & 0 & 0 & 0 & 1
\end{array} \right] \bar{x} = \bar{b}$$

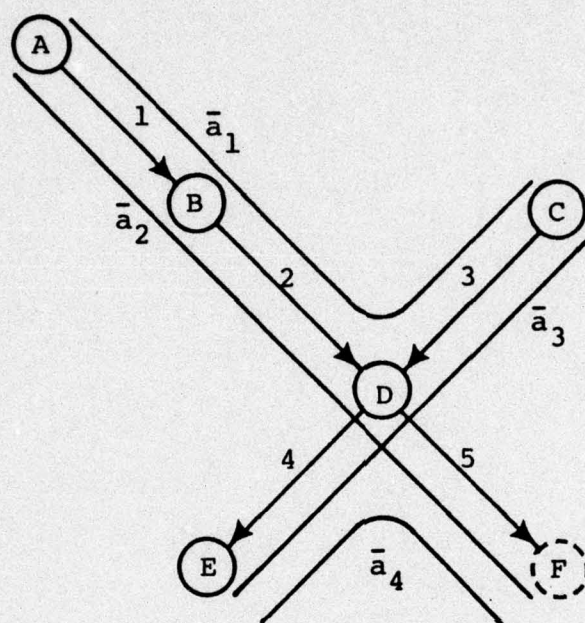$$\bar{x} \geq \bar{0}, \text{ integer}$$

Matrix A may be interpreted as an arc-dipath matrix on the
tree

with tree matrix

$$
T = \begin{bmatrix}
1 & 0 & 0 & 0 & 0 \\
-1 & 1 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 \\
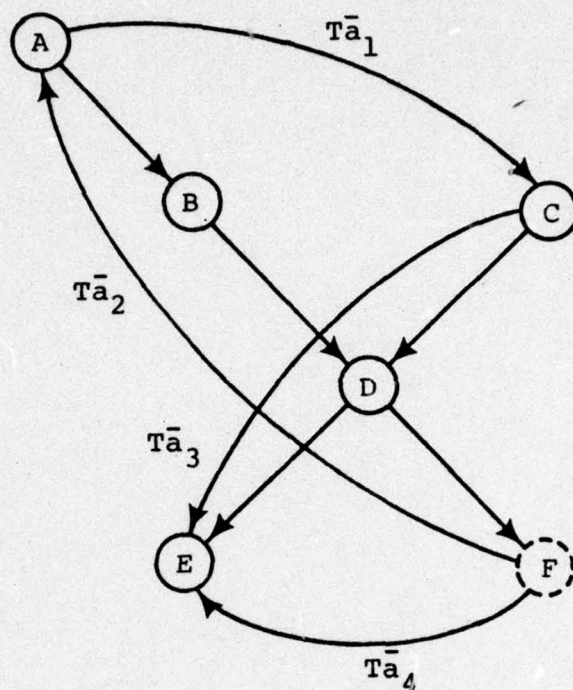0 & -1 & -1 & 1 & 1 \\
0 & 0 & 0 & -1 & 0
\end{bmatrix}
$$

in the following manner,

Thus, transforming the problem via the corresponding tree matrix gives this instance of (3.2),

$$\max \bar{c}\bar{x}$$

$$\text{s.t.} \begin{bmatrix} 1 & -1 & 0 & 0 & | & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & | & -1 & 1 & 0 & 0 & 0 \\ -1 & 0 & 1 & 0 & | & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & | & 0 & -1 & -1 & 1 & 1 \\ 0 & 0 & -1 & -1 & | & 0 & 0 & 0 & -1 & 0 \end{bmatrix} \bar{x} = T\bar{b}$$

$$\bar{x} \geq \bar{0}, \text{ integer}$$

which is a network flow problem on the network

11

It is interesting to note that when A is already a network matrix, then A describes dipaths on the very simple tree given by T = I where I is an mxm identity matrix. This is illustrated in Figure 3.
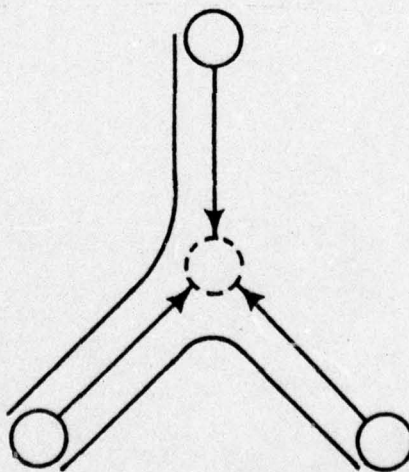

Figure 3: Illustration of Dipaths Described by

$A = \begin{bmatrix} 1 & 0 \\ -1 & -1 \\ 0 & 1 \end{bmatrix}$ on the Tree Given by $T = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$

By this result, recognizing network flow problems is tantamount to recognizing arc-dipath incidence matrices for trees. Iri [14] has developed an $O(m^6)$ algorithm to do just this. In fact his algorithm either transforms a problem to a network flow problem or else, by halting, concludes that such a transformation is not possible. Special cases of this have since been investigated by several authors, e.g., Glover and Klingman [10], Glover et al. [11], Klingman [17].

To some extent recognition of arc-dipath matrices for trees can be accomplished by some easily acquired intuition. Forthnately the idea of "dipaths on a tree" has a certain amount of intuitive appeal that may be lacking in "primitive cutsets of a network" in which Iri's statement of the theorem is phrased. As an example, consider the matrices of consecutive 1's studied by Veinott and Wagner [19]. One such matrix is given in Table 2. It is easy to see (Figure 4) that these matrices describe dipaths on a tree which is itself a dipath and so possess underlying network structure.

$$A = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix}$$

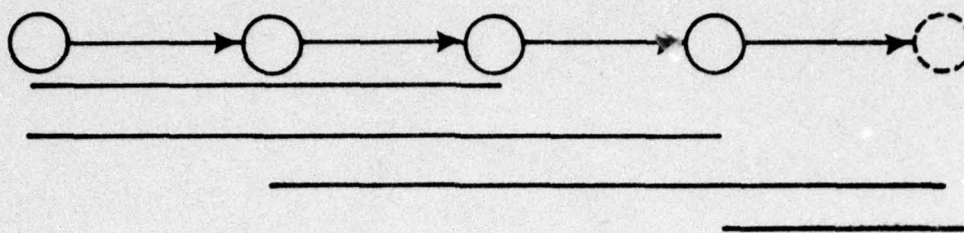Table 2: A Matrix with Consecutive 1's in Columns.

Figure 4: Dipaths Described by the Consecutive 1's
           Matrix in Table 2.

Most of the models to date which have been recognized
as network flow problems arise naturally as either matrices
with a +1 and a -1 in each column or consecutive 1's in
each column. As we have noted, both of these structures
correspond to dipaths on very simple trees. Therefore, it
seems reasonable that gaining additional insight regarding
dipaths on more complex trees might *significantly* expand
the class of problems which can be modeled as network flow
problems.

4. Matchings, Incognito

In the same spirit as Iri, we now ask when an integer
linear program is equivalent to a maximum b-matching problem
and so solvable with guaranteed efficiency. Again, let A be
a 0, ± 1 matrix and $\bar{b}$, $\bar{c}$ integer-valued vectors. Consider the
problem

$$\max \bar{c}\bar{x}$$
$$\text{s.t. } A\bar{x} \leq \bar{b} \qquad\qquad (4.1)$$
$$\bar{x} \geq \bar{0}, \text{ integer}$$

and the equivalent, transformed version

$$\max \bar{c}\bar{x}$$
$$\text{s.t. } [TA, T]\bar{x} = T\bar{b}, \text{ T nonsingular} \qquad (4.2)$$
$$\bar{x} \geq \bar{0}, \text{ integer}$$

14

Now by Edmonds [5], the basic matching algorithm is applicable to (4.2) when the following are satisfied:

Requirement 1: All data are integer-valued, and

Requirement 2: The constraint matrix is the node-arc incidence matrix of a graph; that is, no column may have more than two nonzero elements, each a +1. We refer to such a matrix as a graph matrix.

An expanded form of the matching algorithm [7] will in fact apply even when Requirement 2 is relaxed to

Requirement 2': For any column $\bar{c}_j$ of the constraint matrix, $\sum_i |c_{ij}| \leq 2$. Thus, each column contains all 0's except for one +1, or one -1, or one +2, or one -2, or two +1's, or two -1's, or a +1 and a -1.

We restrict ourselves to the simpler case in order to more clearly present the ideas involved. Extension to the more general form of the algorithm is straightforward and is indicated in Appendix 2.

The derivation of conditions is analogous to that of Section 3. In this case, it is clear that T must be a graph matrix.

Lemma 4.1: A graph matrix is nonsingular iff it is the graph matrix of a bloom, or the full row-rank graph matrix of a tree.

Proof: Without loss of generality we may assume T is connected, since otherwise problem (4.2) is separable. Then since T has at most m arcs and at least m nodes, T contains at most one cycle. But this cannot be an even cycle. If it were, with arcs (columns of matrix T) $\bar{t}_1$, $\bar{t}_2$, ..., $\bar{t}_{2k}$, then $\sum_{j=1}^{2k} (-1)^j \bar{t}_j = \bar{0}$, a contradiction to T nonsingular. Thus, T

15

contains either no cycles and is a tree, or else contains
exactly one odd cycle, and is a bloom.

To see that such matrices are indeed nonsingular, see
Appendix 1 for a simple inversion scheme.

<div align="right">Q.E.D.</div>

Note that when the more general definition of a matching
matrix is used (Requirement 2'), an appropriately more
general version of Lemma 4.1 ensues (see Appendix 2).

Again, as in Section 3 , we may interpret each col-
umn $\bar{a}_j$ as picking out and orienting a set of arcs in the
tree/bloom T. However, to maintain the graphical metaphor,
as in Edmonds et al. [7], we interpret an undirected arc,
i.e., one corresponding to a column with two +1's, as having
a tail on each end, as in

Such an arc will be called "inner-directed." Thus, when arc
i is reoriented by $a_{ij} = -1$ for some j, it has a head on each
end, and corresponds to a column with two -1's, as in

Such an arc will be called "outer-directed."

16

As before, $T\bar{a}_j$ gives the net degrees of each of the nodes of T (except for the root node when T is a tree) in the set of arcs $\bar{a}_j$. Thus, the reasoning of the previous section establishes that the arcs of $\bar{a}_j$ form a (not necessarily simple) path in T. To discover the additional special structure of $\bar{a}_j$, consider the following,

Case 1: T a tree

Assume the path $\bar{a}_j$ does not contain the root node. Then since the net degrees of nodes interior to the path must be zero, $\bar{a}_j$ must orient the arcs of the path to form an <u>alternating path</u>, that is, one in which successive arcs are alternately inner- and outer-directed, as in

+1   0   0   +1 : net degrees of nodes in $\bar{a}_j$

: alternating path $\bar{a}_j$
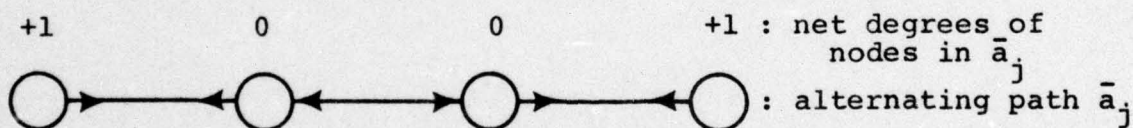
Furthermore, so that the net degrees of the terminal nodes are +1, the terminal arcs must be inner-directed.

If the path $\bar{a}_j$ does contain the root node, then since $\bar{a}_j$ is a simple path, it is divided into two subpaths by the root node. But by the previous argument, each of these subpaths must be alternating and have terminal arcs which are inner-directed as in

: path $\bar{a}_j$

+1   0   root node   0   +1 : net degrees of nodes in $\bar{a}_j$
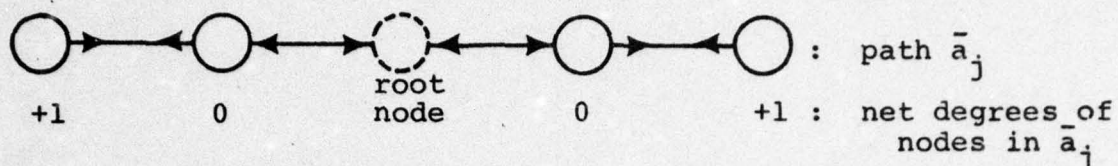
Therefore, $\bar{a}$ must describe a path with inner-directed arcs such that each path is either

       (i) an alternating path, or else

       (ii) a path divided by the root node into two alternating subpaths.

Case 2:   T a bloom

The previous argument shows that $\bar{a}_j$ must orient arcs to form an alternating path in T with terminal arcs that are inner directed. However, since a bloom contains a cycle C, an extra stipulation is needed: the alternating path cannot be only the cycle C, since then one node would have net degree +2, as in
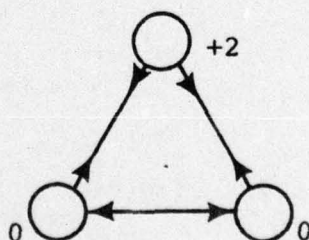
Thus $\bar{a}_j$ must describe an alternating path on the bloom T with inner-directed terminal arcs and with distinct terminal nodes.

Finally, we observe that since T is an integer matrix, all of the data of problem (4.2) are integer so that Requirement 2 is satisfied.

18

Therefore, we have

Theorem 4.1: Problem (4.2) is a matching problem (restricted version) iff there exists some tree or bloom T for which A is an arc-path incidence matrix such that each path
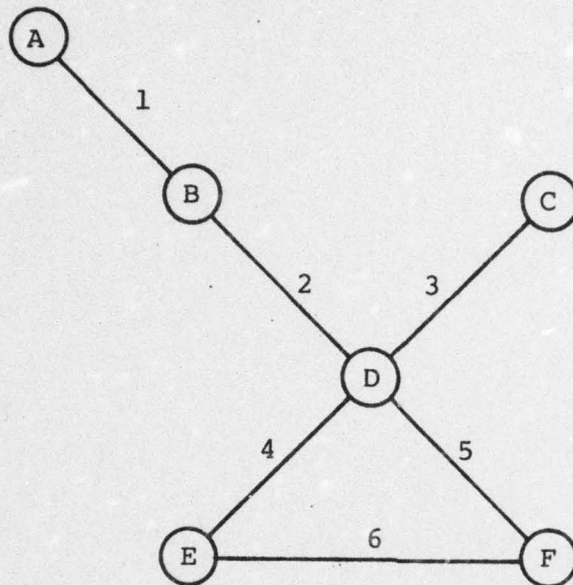
    (i) has inner-directed terminal arcs, and

   (ii) has distinct terminal nodes, and

  (iii) is either alternating or else is divided by the
               root node into two alternating subpaths.

Proof: Since each step of the argument can be made "if-and-only-if," the conclusion follows.

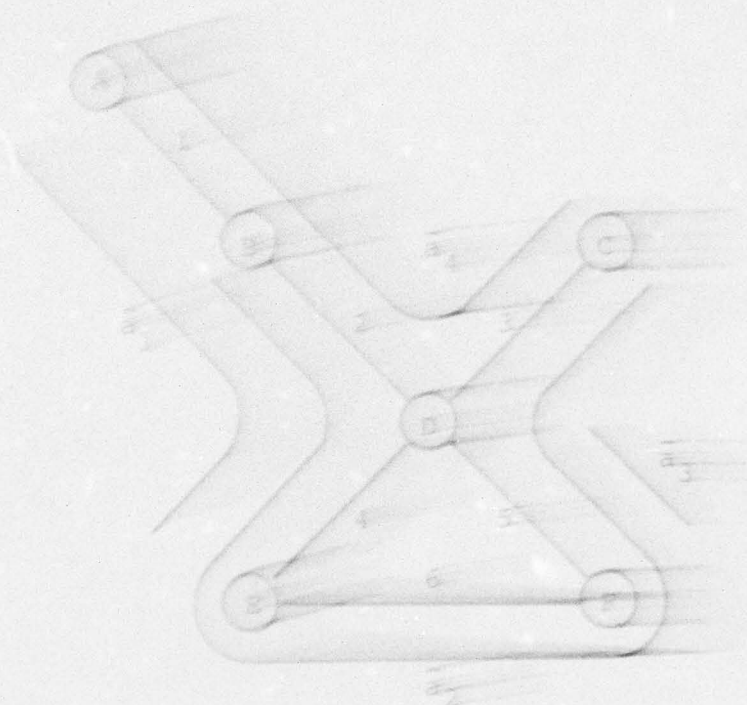Consider the following instance of problem (4.1),

$$\max \bar{c}\bar{x}$$

$$\text{s.t.} \quad
\begin{array}{c}
1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6
\end{array}
\left[
\begin{array}{cccc|cccccc}
1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\
-1 & 1 & 0 & -1 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\
1 & -1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1
\end{array}
\right]
\bar{x} = \bar{b}$$

$$\bar{x} \geq \bar{0}, \text{ integer}$$

Matrix A may be realized as an arc-path incidence matrix on the bloom

with bloom matrix

$$
T = \begin{array}{c|cccccc}
 & 1 & 2 & 3 & 4 & 5 & 6 \\
A & 1 & 0 & 0 & 0 & 0 & 0 \\
B & 1 & 1 & 0 & 0 & 0 & 0 \\
C & 0 & 0 & 0 & 0 & 0 & 1 \\
D & 0 & 1 & 1 & 0 & 1 & 1 \\
E & 0 & 0 & 1 & 1 & 0 & 0 \\
F & 0 & 0 & 0 & 1 & 1 & 0 \\
\end{array}
$$

where all paths of x meet the conditions of the theorem.

Thus we may transform the problem by T to

$$\max cx$$

A	1	0	0	1	1	0	0	0	0	0	
B	0	1	0	0	1	1	0	0	0	0	
C	0	1	0	1	0	0	0	0	0	1	
D	0	0	1	0	0	1	1	0	1	1	
E	1	0	0	0	0	0	1	1	0	0	
F	0	0	1	0	0	0	0	1	1	0	

$$x = Tb$$

$$x \geq 0, \text{ integer}$$

which is a maximum b-matching problem on the graph
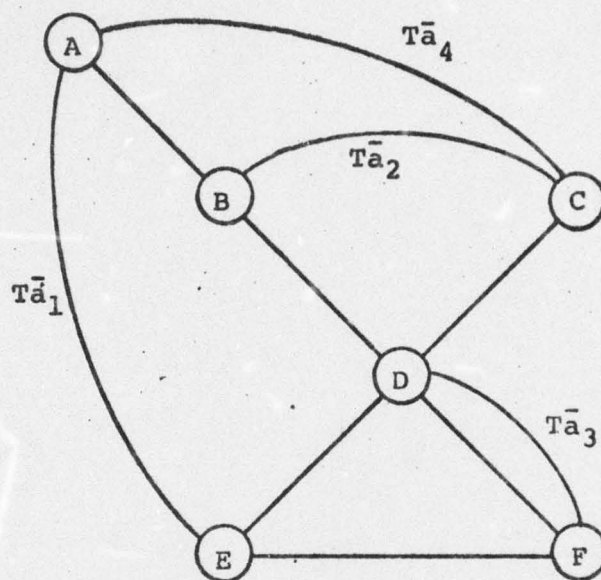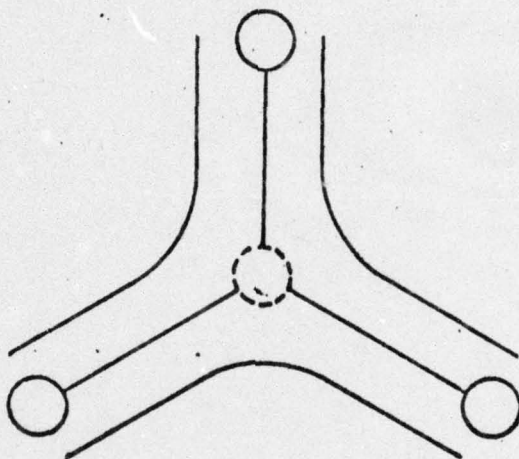

Also note that for A already a matching matrix, A
must describe paths on the tree given by T = I.

As an example consider

$$A = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix}$$

Then on the tree given by $T = I$, A describes paths
with inner-directed terminal arcs such that the paths are
divided by the root node into two alternating subpaths, as in

Finally, we observe that conditions (i) and (ii) are
necessary only for the restricted version of the matching
algorithm. Any problem which meets all the conditions of
the theorem but not (i) and (ii) is still solvable via the
generalized matching algorithm.

5.   Additional Considerations

As a practical matter in identifying network flow or
matching problems, the case of equality constraints must
be considered. Suppose the original problem was of the
form

$$\max \ \bar{c}\bar{x}$$
$$\text{s.t.} \ \hat{A}\bar{x} = \bar{b} \qquad\qquad (5.1)$$
$$\bar{x} \geq \bar{0}$$

where $\hat{A}$ is a full row-rank matrix of dimensions m x n (m < n). Let B be a basis of $\hat{A}$ so that $\hat{A}$ may be partitioned as $\hat{A} = [B,N]$. Then transforming (5.1) by B^{-1} yields the equivalent problem

$$\max \bar{c}\bar{x}$$
$$\text{s.t. } [B^{-1}N,I]\bar{x} = B^{-1}\bar{b} \qquad (5.2)$$
$$\bar{x} \geq \bar{0}$$

to which Theorems 3.1 and 4.1 apply, with $A = B^{-1}N$.

Note that, if transformable to a <u>network</u> matrix, A must be totally unimodular since TA must be totally unimodular [15]. Furthermore, by the correspondence of extreme points, $\hat{A}$ must have all bases unimodular [1].

APPENDIX 1

INVERTING A NONSINGULAR GRAPH MATRIX

Inverting a nonsingular graph matrix is similar to inverting a nonsingular network matrix [16]. We consider the two cases:

(i) Let T be the m x m full row-rank graph matrix of a tree and let r be the root (implicit) node. Then since T is a tree, for any node i there is a unique path P_i to node r. Let $\bar{p}_i$ be the 0, ±1 vector that picks out the arcs in P_i and orients them so that they are alternating, with the arc incident at node i inner-directed. Then in $\bar{p}_i$ the net degree of node i is +1 and that of all other nodes is 0 (see Example A1.1), so that $T\bar{p}_i = \bar{e}_i$. Then if P is the m x m matrix whose ith colum is $\bar{p}_i$, we have TP = I, so that $P = T^{-1}$.

As an illustration consider

$$
T =
\begin{array}{c}
 \\
A \\
B \\
C \\
D \\
E
\end{array}
\begin{array}{c}
\begin{array}{ccccc}
1 & 2 & 3 & 4 & 5
\end{array} \\
\left[
\begin{array}{ccccc}
1 & 0 & 0 & 0 & 0 \\
1 & 1 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 \\
0 & 1 & 1 & 1 & 1 \\
0 & 0 & 0 & 1 & 0
\end{array}
\right]
\end{array}
$$

corresponding to the tree
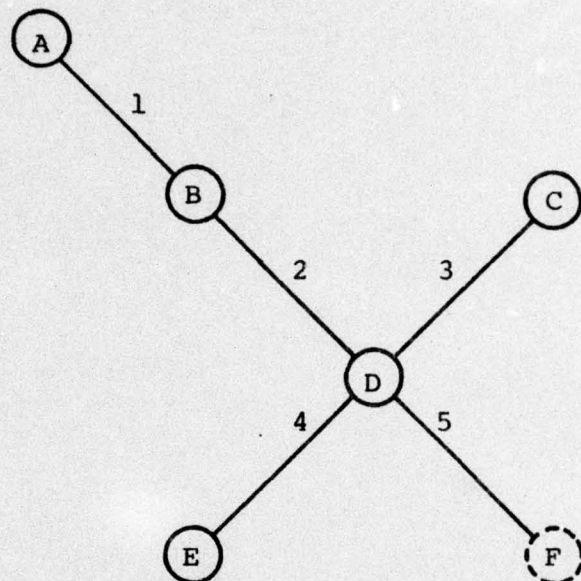
with root node F.

Then $\bar{p}_a = [+1, -1, 0, 0, +1]^t$, corresponding to the alternating path
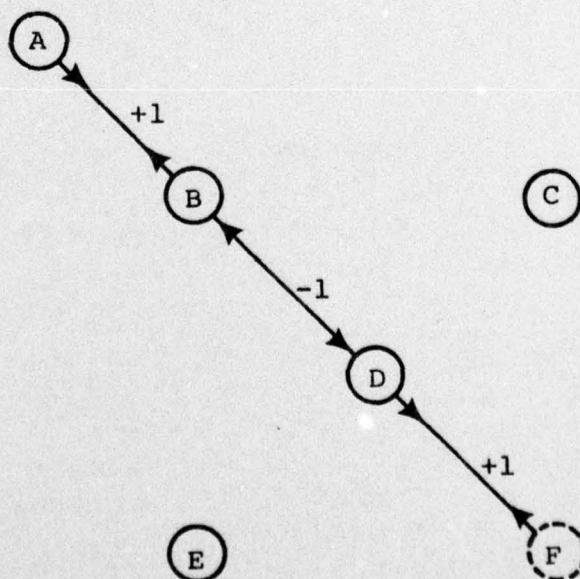
in which only node A has net degree $\neq 0$, so that $T\bar{p}_a = [1, 0, 0, 0, 0]^t$.

Continuing, we compute

$$
T^{-1} = \begin{array}{c} \begin{array}{ccccc} \bar{p}_A & \bar{p}_B & \bar{p}_C & \bar{p}_D & \bar{p}_E \end{array} \\ \left[\begin{array}{ccccc} 1 & 0 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 1 & -1 & -1 & 1 & -1 \end{array} \right] \end{array}
$$

(ii) Suppose T is the m x m graph matrix of a bloom. In this case the odd cycle plays a role similar to that of a root node. For any node i there is a unique path P_i to the odd cycle C. Now let $\bar{p}_i$ be the vector that identifies arcs by entries ± 1 for arcs in P_i, $\pm 1/2$ for arcs in C, and 0 for all other arcs of T. Furthermore let the signs of the entries in $\bar{p}_i$ be chosen to orient the arcs of $P_i \cup C$ to form an alternating path with the arc incident at node i inner-directed. Then in $\bar{p}_i$ the net degree of node i is $+1$ and that of all other nodes is 0 (see Example A1.2), so that $T\bar{p}_i = \bar{e}_i$. Then if P is the m x m matrix whose ith column is $\bar{p}_i$, $TP = I$ so that $P = T^{-1}$.

27

As an illustration consider

$$
T = \quad
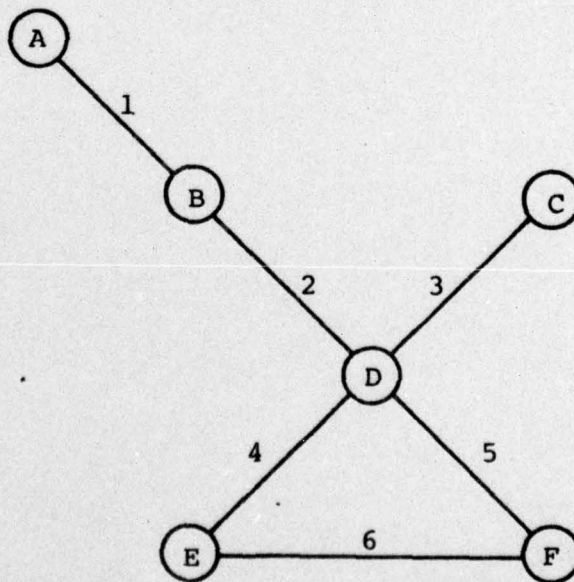\begin{array}{c}
 \\ A \\ B \\ C \\ D \\ E \\ F
\end{array}
\begin{array}{cccccc}
1 & 2 & 3 & 4 & 5 & 6 \\
\left[\begin{array}{cccccc}
1 & 0 & 0 & 0 & 0 & 0 \\
1 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 \\
0 & 1 & 1 & 1 & 1 & 0 \\
0 & 0 & 0 & 1 & 0 & 1 \\
0 & 0 & 0 & 0 & 1 & 1
\end{array}\right]
\end{array}
$$

corresponding to the bloom

28

Then $\bar{p}_a = [+1, -1, 0, +\frac{1}{2}, +\frac{1}{2}, -\frac{1}{2}]$, corresponding to the weighted alternating path
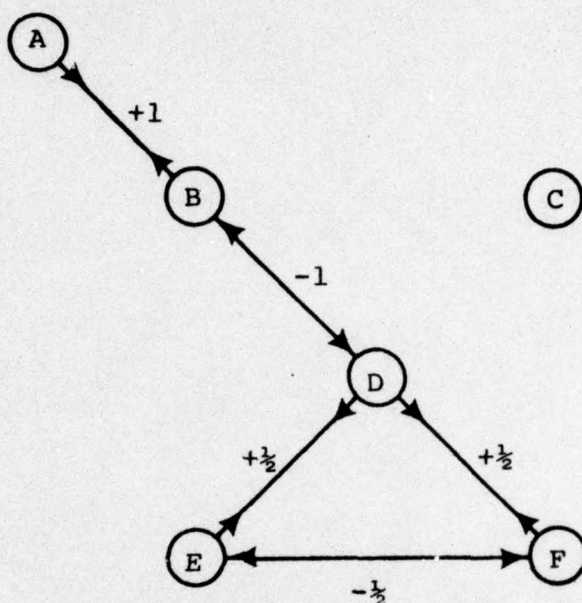
in which only node A has net degree $\neq 0$, so that $T\bar{p}_a = [1, 0, 0, 0, 0, 0]^t$.

Continuing, we compute

$$
T^{-1} = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{array}
\begin{array}{cccccc}
\bar{p}_A & \bar{p}_B & \bar{p}_C & \bar{p}_D & \bar{p}_E & \bar{p}_F \\
\left[\begin{array}{cccccc}
1 & 0 & 0 & 0 & 0 & 0 \\
-1 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 \\
\frac{1}{2} & -\frac{1}{2} & -\frac{1}{2} & \frac{1}{2} & \frac{1}{2} & -\frac{1}{2} \\
\frac{1}{2} & -\frac{1}{2} & -\frac{1}{2} & \frac{1}{2} & -\frac{1}{2} & \frac{1}{2} \\
-\frac{1}{2} & \frac{1}{2} & \frac{1}{2} & -\frac{1}{2} & \frac{1}{2} & \frac{1}{2}
\end{array}\right]
\end{array}
$$

APPENDIX 2

RECOGNIZING A GENERAL MATCHING PROBLEM

The most general form of a matching matrix, that of Requirement 2', may be interpreted as the node-arc incidence matrix of a so-called "bi-directed" graph [7]. According to this scheme, the rows of the matrix correspond to nodes of the graph, and the columns to arcs, where

(i) a column with one +1 and one -1 corresponds to a directed arc, i.e., one with a tail on one end and a head on the other.

(ii) a column with two +1's (-1's) corresponds to an inner- (outer-) directed arc, i.e., one with a tail (head) at each end.

(iii) a column with one +1 (-1) corresponds to a "spike," i.e., an arc with a tail (head) at the node end.

(iv) a column with one +2 (-2) corresponds to a loop, i.e., an arc with both ends incident at the same node, with a tail (head) at each end. Such arcs are inner- (outer-) directed.

With this enlarged metaphor, it is possible to extend Lemma 4.1 to

Lemma 4.1': A bi-directed graph matrix is nonsingular iff it is the full row-rank matrix of a bi-directed graph with at

30

most one cycle, that cycle having an odd number of undirected (i.e., inner- or outer-directed) arcs.

We will call the graph of Lemma 4.1' a <u>bi-directed bloom</u>. Then the same sort of argument as Section 4 establishes <u>Theorem 4.1'</u>: Problem (4.2) is a matching problem (generalized version) <u>iff</u> there exists some bi-directed bloom for which A describes paths such that the net degree of any interior node (other than possibly a root node) is 0.

BIBLIOGRAPHY

1. Bartholdi, J. J. and H. D. Ratliff, "Unimodular and Totally Unimodular Matrices," Research Report No. 76-21, August 1976, Industrial and Systems Engineering Department, University of Florida, Gainesville, Florida.

2. Bratley, P., M. Florian, and P. Robillard, "Scheduling with Earliest Start and Due Date Constraints," Naval Research Logistics Quarterly, Vol. 18, No. 4, December 1971, pp. 511-19.

3. Dantzig, G. B. and D. R. Fulkerson, "Minimizing the Number of Tankers to Meet a Fixed Schedule," Naval Research Logistics Quarterly, Vol. 1, 1964, pp. 217-22.

4. de Werra, D., "On Some Combinatorial Problems Arising in Scheduling," Working Paper No. 20, N.D., Department of Management Science, University of Waterloo, Waterloo, Ontario.

5. Dorsey, R. C., T. J. Hodgson, and H. D. Ratliff, "A Network Approach to a Multi-Facility, Multi-Product Production Scheduling Problem without Back-Ordering," Management Science, Vol. 21, No. 7, March 1975, pp. 813-22.

6. Edmonds, J., "Paths, Trees, and Flowers," Canadian Journal of Mathematics, Vol. 17, 1965, pp. 449-67.

7. Edmonds, J., E. L. Johnson, and S. C. Lockhart, "Blossom I: A Computer Code for the Matching Problem," from a series of instructional lectures entitled "Optimum Matching and Polyhedral Combinatorics," given by J. Edmonds and W. Pulleyblank at the Johns Hopkins University, Baltimore, Maryland, June 2-6, 1975.

8. Ford, L. R. and D. R. Fulkerson, Flows in Networks, Princeton University Press, Princeton, New Jersey, 1962.

9. Fujii, M., T. Kasami, and K. Ninomiya, "Optimal Sequencing of Two Equivalent Processors," SIAM Journal of Applied Mathematics, Vol. 17, No. 4, July 1969, pp. 784-9.

10. Glover, F. and D. Klingman, "On the Equivalence of Some Generalized Network Problems to Pure Network Problems," Mathematical Programming, Vol. 4, No. 3, June 1973, pp. 269-78.

11. Glover, F., D. Klingman, and G. T. Ross, "Finding Equivalent Transportation Formulations for Constrained Transportation Problems," Research Report CS 107, Center for Cybernetic Studies, University of Texas, Austin, Texas, January 1973.

12. Harary, F., Graph Theory, Addison-Wesley Publishing Company, Inc., Reading, Massachusetts, 1969.

13. Iri, M., "A Criterion for the Reducibility of a Linear Programming Problem to a Linear Network Flow Problem," RAAG Research Notes, Third Series, No. 98, February 1966.

14. Iri, M., "On the Synthesis of Loop and Cutset Matrices and the Related Problems," RAAG Memoirs, Vol. 4, 1968, pp. 4-38.

15. Iri, M., Network Flow, Transportation, and Scheduling, Academic Press, New York, New York, 1969.

16. Johnson, E., "Programming in Networks and Graphs," ORC 65-1, Operations Research Center, University of California at Berkeley, Berkeley, California, January 1965.

17. Klingman, D., "Equivalent Network Formulations for Constrained Networks," Management Science, Vol. 23, No. 7, March 1977, pp. 737-44.

18. Love, R. R., Jr. and R. R. Vermuganti, "Single-Plant Multi-Commodity Production and Distribution Scheduling with Capacity and Changeover Restrictions," Working Paper, 1975.

19. Veinott, A. G., Jr. and H. M. Wagner, "Optimal Capacity Scheduling - I and II," Operations Research, Vol. 10, No. 4, 1962, pp. 518-46.